**Transaction Isolation Levels**

Among the ACID properties of transactions, SQL Server never compromises the atomicity, consistency, and durability requirements of a database transaction. However, the degree of isolation can vary for readers depending on the settings that their session applies.

During the time a transaction is changing some data, SQL Server never allows that data to be changed by any other transaction until the first transaction finishes, nor can your transaction change any data that other transactions are changing until they finish. Therefore, some blocking and deadlocking is always possible when transactions change data. Writers always block writers, and exclusive locks in one transaction are never compatible with exclusive locks in another.

But blocking and deadlocking can be increased or reduced based on varying the degree of isolation of the transaction ACID properties. SQL Server allows your transaction to read other transactions’ data or allows data to be changed by other transactions that the current transaction only reads, based on the setting of what is called the transaction isolation level.

The most commonly used isolation levels are:

* **READ COMMITTED** This is the default isolation level. All readers in that session will only read data changes that have been committed. So all the SELECT statements will attempt to acquire shared locks, and any underlying data resources that are being changed by a different session, and therefore have exclusive locks, will block the READ COMMITTED session.
* **READ UNCOMMMITED** This isolation level allows readers to read uncommitted data. This setting removes the shared locks taken by SELECT statements so that readers are no longer blocked by writers. However, the results of a SELECT statement could read uncommitted data that was changed during a transaction and then later rolled back to its initial state. This is called reading dirty data.
* **READ COMMITTED SNAPSHOT** This is actually not a new isolation level; it is an optional way of using the default READ COMMITTED isolation level, the default isolation level in Windows Azure SQL Database. This isolation level has the following traits:
  + Often abbreviated as RCSI, it uses tempdb to store original versions of changed data. These versions are only stored as long as they are needed to allow readers (that is, SELECT statements) to read underlying data in its original state. As a result, SELECT statements no longer need shared locks on the underlying resource while only reading (originally) committed data.
  + The READ COMMITTED SNAPSHOT option is set at the database level and is a persistent database property.
  + RCSI is not a separate isolation level; it is only a different way of implementing READ COMMITTED, preventing writers from blocking readers.
  + RCSI is the default isolation level for Windows Azure SQL Database
* **REPEATABLE READ** This isolation level, also set per session, guarantees that whatever data is read in a transaction can be re-read later in the transaction. Updates and deletes of rows already selected are prevented. As a result, shared locks are kept until the end of a transaction. However, the transaction may see new rows added after its first read; this is called a phantom read.
* **SNAPSHOT**This isolation level also uses row versioning in tempdb (as does RCSI). It is enabled as a persistent database property and then set per transaction. A transaction using the SNAPSHOT isolation level will be able to repeat any reads, and it will not see any phantom reads. New rows may be added to a table, but the transaction will not see them. Because it uses row versioning, the SNAPSHOT isolation level does not

require shared locks on the underlying data.

* **SERIALIZABLE**This isolation level is the strongest level and is set per session. At this

level, all reads are repeatable and new rows are not allowed in the underlying tables

that would satisfy the conditions of the SELECT statements in the transaction